CLAIMS

1. A computer program product for rule-based random irritation of a UUT model in a simulation of a UUT, the UUT model being a software model of the UUT, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

an extractor program code for extracting one or more inputs and one or more outputs of the UUT;

a pattern generator program code for generating data patterns to be applied to the UUT model in accordance with a rules list and information on the one or more inputs and the one or more outputs of the UUT, the rules list providing restrictions or encouragements on how the data patterns are to be applied to the UUT model;

a simulator program code for applying the data patterns to the UUT model; and

a software environment program code for interfacing communications between the extractor program code, the pattern generator program code, and the simulator program code.

2 The computer program product of Claim 1, wherein the extractor program code generates a first computer file containing information on the one or more inputs and the one or more outputs of the UUT.

3. The computer program product of Claim 1, wherein the software environment program code calls the extractor program code and provides the extractor program code with the paths to a plurality of computer files that describe the logic of the UUT, and wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more outputs of the UUT.

4. The computer program product of Claim 1, wherein the software environment program code calls the extractor and provides the extractor with the paths to a plurality of

computer files that describe the logic of the UUT, wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more output of the UUT, and wherein the plurality of computer files are in hardware description language.

5. The computer program product of Claim 1, wherein the simulation comprises one or more events, and for each event, the software environment program code prompts the pattern generator program code to generate a data pattern, and the software environment program code interprets the data pattern and passes the data pattern so interpreted to the simulator.

6. The computer program product of Claim 1, wherein the simulation comprises one or more events, and for each event, the software environment program code prompts the pattern generator program code to generate a data pattern, and the software environment program code interprets the data pattern and passes the data pattern so interpreted to the simulator, and wherein, when prompted by the software environment program code, for each rule listed in the rules list, the pattern generator assigns a value dictated by the rule to nets affected by the rule, and, for the remaining nets not associated with a rule in the rules list, creates random values.

7. A computer program product for rule-based random irritation of a UUT model, the UUT model being a software model of a UUT, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

an extractor program code for extracting one or more inputs and one or more outputs of the UUT;

a pattern generator program code for generating data patterns to be applied to the UUT model in accordance with a rules list and information on the one or more inputs and the one or more outputs of the UUT, the rules list providing restrictions or encouragements on how the data patterns are to be applied to the UUT model; and

a simulator program code for applying the data patterns to the UUT model, the simulator

program code includes the pattern generator program code.

8. The computer program product of Claim 7, wherein the extractor program code generates a first computer file containing information on the one or more inputs and the one or more outputs of the UUT.

9. The computer program product of Claim 7, wherein the simulator program code calls the extractor program code and provides the extractor program code with the paths to a plurality of computer files that describe the logic of the UUT, and wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more outputs of the UUT.

10. The computer program product of Claim 7, wherein the simulator program code calls the extractor program code and provides the extractor program code with the paths to a plurality of computer files that describe the logic of the UUT, and wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more outputs of the UUT, and wherein the plurality of computer files are in hardware description language.

11. The computer program product of Claim 7, wherein the simulation comprises one or more events, and for each event, the simulator program code prompts the pattern generator program code to generate a data pattern, and the pattern generator program code passes the data pattern to the simulator program code in the language of the simulator.

12. The computer program product of Claim 7, wherein the simulation comprises one or more events, and for each event, the simulator program code prompts the pattern generator program code to generate a data pattern, and the pattern generator program code passes the data

pattern to the simulator program code in the language of the simulator program code, and wherein, when prompted by the simulator program code, for each rule listed in the rules list, the pattern generator program code assigns a value dictated by the rule to nets affected by the rule, and, for the remaining nets not associated with a rule in the rules list, creates random values.

13. A method for rule-based random irritation of a UUT model in a simulation of a UUT, the UUT model being a software model of the UUT, the simulation having one or more events, the method comprising the steps of:

generating a rules list, the rules list providing information on how the data patterns are to be applied to the UUT model;

generating a data pattern, for an event of the simulation, in accordance with the rules list and information on one or more inputs and one or more outputs of the UUT;

performing each event by applying the respective data pattern to the UUT model; and

determining whether all events of the simulation are performed.

14. The method of Claim 13, further comprising the steps of:

ending the simulation, if all events of the simulation are performed; and

repeating the steps of generating a data pattern, performing each event, and determining whether all events of the simulation are performed, if all events of the simulation are not performed.

15. The method of Claim 13, further comprising the step of generating an I/O file containing information on one or more inputs and one or more outputs of the UUT.

16. The method of Claim 13, further comprising the steps of:

calling an extractor program code for extracting one or more inputs and one or more outputs of the UUT and generating a first computer file containing information on the one or

more inputs and the one or more output of the UUT; and

providing the extractor program code with the paths to a plurality of computer files that describe the logic of the UUT, and wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more outputs of the UUT.

17. The method of Claim 13, further comprising the steps of:

prompting to generate the data pattern;

interpreting the data pattern; and

passing the data pattern to a simulator performing the simulation.

18. The method of Claim 13, further comprising the steps of:

prompting to generate the data pattern;

interpreting the data pattern;

passing the data pattern to a simulator performing the simulation;

assigning, for each rule contained in the rules list, a value to nets affected by the rule, and, for the remaining nets not associated with a rule in the rules list, creating random values.

19. An apparatus for rule-based random irritation of a UUT model in a simulation of a UUT, the UUT model being a software model of the UUT, the apparatus comprising:

means for generating a rules list, the rules list providing information on how the data patterns are to be applied to the UUT model;

means for generating a data pattern, for an event of the simulation, in accordance with the rules list and information on one or more inputs and one or more outputs of the UUT;

means for performing each event by applying the respective data pattern to the UUT model; and

means for determining whether all events of the simulation are performed.

20. The apparatus of Claim 19, further comprising means for generating an I/O file containing information on one or more inputs and one or more outputs of the UUT

21. The apparatus of Claim 19, further comprising:

means for calling an extractor program code for extracting one or more inputs and one or more outputs of the UUT and generating a first computer file containing information on the one or more inputs and the one or more output of the UUT; and

means for providing the extractor program code with the paths to a plurality of computer files that describe the logic of the UUT, and wherein the extractor program code uses the plurality of the computer files to extract the one or more inputs and the one or more outputs of the UUT.

22. The apparatus of Claim 19, further comprising:

means for prompting to generate the data pattern;

means for interpreting the data pattern; and

means for passing the data pattern to a simulator performing the simulation

23. The apparatus of Claim 19, further comprising:

means for prompting to generate the data pattern;

means for interpreting the data pattern;

means for passing the data pattern to a simulator performing the simulation;

means for assigning, for each rule contained in the rules list, a value to nets affected by the rule, and, for the remaining nets not associated with a rule in the rules list, creating random values.